Fred Truck's ArtEngine :
a case study in the problematics of software art

Simon Penny
Mediamatic Holland, V5#3, 1990

I recently took Fred Trucks' ArtEngine for a test drive. The ArtEngine is: "a robot artwork that makes art"(sic).(1) It attempts to model the interaction of visual and associative memory functions in human creative thought. To do this it utilises artificial intelligence techniques and aspects of expert systems design.

ArtEngine is menu driven at the front end, but attempts a more complex interactivity. It requires the operator to create original lists of objects in the Scheme programming language. Thus it immediately engages the question of 'depth' of interactivity: how much learning should be required of the user before they can usefully interact, and the corollary: how rich can the harvest of this interaction be made to be.

To elucidate: ArtEngine is not one of those familiar software packages which simulates conventional forms on a video surface or is formulated as the simulation of a conventional tool to aid the simulation of such forms. Into this category one can place almost all the 'user friendly' graphics packages. These are of course subject to the lowest common denominator enforcement of the restriction 'user friendly' which is a major impediment to their complexity. One drives these programs in the same sense that one drives a car: their structure is set.

ArtEngine thus engages two more of the basic computer art debates, firstly: " Is it possible to make worthwhile art with proprietary software packages or do all computer artists have to be programmers?" and secondly: "Is it possible to map the intuitive/associative/inductive methodology of (more recent) traditional art practice on to the mechanistic quality of computer logic to any useful end?" There are those who argue that truly creative work with computers can only be achieved through custom programming. "There is no such thing as a general purpose machine", or to put it another way, the more generally useful a machine is the less specifically useful it is. Longtime artificial intelligence artist Harold Cohen makes the distinction between the 'expert system' and his 'experts' system'. (2)

The first half of the manual to ArtEngine is an engaging interdisciplinary text titled: "ArtEngine (a symbolic machine): Portrait of the artist as an information processor". Old enough to have seen Jasia Riechardts' landmark exhibition 'Cybernetic Serendipity" in 1969, and smart enough to have been lastingly impressed, Truck discusses such diverse subjects as programming languages, communications theory, Boolean algebra and the function of the self portrait from Titian to Warhol.

What then is the ArtEngine? It has a professed dual identity as an artwork, and as a tool for making art. We should not expect exclusive logic to apply; in this new medium it may be possible to simultaneously be an artwork and a tool. As truck explains in the manual, one function, in terms of the program, is the other one backwards. Further consideration suggests that the work is neither an artwork nor a tool for making art, but that the software and manual function as an interactive hypermedia essay on the nature of machine intelligence and its relationship to art practice.

The dual identity of the ArtEngine is an example of the level of conceptual punning that is built into the structure of the work. Truck mentions early on in the manual, that the operator (player?, artist?, interacter?, there is a language lag here) will find more esthetic value in the implications of the program and the concepts embodied, than in the graphical and textual output. While this is faintly disappointing at the outset, it does ground the user in post-object esthetics as a critical vantage point. This is singularly appropriate to software artwork which, by its nature, must be cerebral, multidimensional and concerned with disembodied conceptual engineering.(3)

ArtEngine adopts a position midway between a user-friendly (read: foolproof) application and a complex programming task. It is easy to see the rationale for this. A user- friendly front end (Hypercard) serves to reassure the faint hearted. One hits the programming level soon enough. The 'user friendly front end' is a pragmatic strategy for easing the interlocutor into a complex meta-topography of ideas, a fluid universe Truck has designed as an interactive artwork. However, in order to derive the pleasure of fluent complex interactivity, one needs to be familiar with such arcane corners of the Mac system as the Quickdraw algorithms. Some familiarity with the 'scheme' programming language is also very useful.

None of us expect to be able to drive a new piece of software just by loading it, yet we expect that art works are that accessible. We forget that we have a lifetime of cultural education behind us that makes that artwork accessible, we've read the manual. Software art is new, as software is new, and potential users are not aculturated.

Which leads us back to the vexing question of ‘depth’ of interactivity. Any artist preparing interactive work confronts the design problem of effectively leading the user through the universe of the work in such a way that the process is neither pedantically simple-minded nor mind-bogglingly complex. Any new piece of software demands a learning period like any other skill, an investment of time, in order to exploit it fully. User-friendly applications are not powerful; specialized programs require enskilling, or they speak only to a skilled elite. The horns of the software art dilemma are the horns of the software dilemma as a whole: There is no such thing as a truly powerful user friendly application. How then do we construct a model for rewarding interactivity in a user friendly box?

The ArtEngine purports to make art, it purports to be modelled on Trucks' own strategies for artmaking. The particular strategy embodied in ArtEngine is basically a dialectical process of deriving a synthesis of two juxtaposed sets of ideas. The program is so constructed that the 'sets of ideas' manifest as graphic information and text simultaneously, at the program level the same information gives rise to both. One creates the two ideas and the machine divides and multiplies, derives a synthesis of the two.

If this is the artistic process at work then it is once again faintly disappointing. For if the ArtEngine thinks that artmaking is simply a process of mixing and matching then it confirms our suspicions that artmaking is a very complex process and machines have a long way to go to emulate it.

Art making in late C20th western society may be the extreme case of synthetic (as opposed to analytical) intellectual activity. It is a modernist cliche', and the modus operandi of avant-gardism, that art continually 'expands the boundaries', 'transgresses traditional forms' etc. It is exactly this that the computer is currently unable to do, because it is confined to a rule-bound world.(4) This is a disability of machines, it is generally assumed that the human intellect is not subject to such limitations. Further, art making is an extremely 'self conscious' activity. The artist has some conception of what s/he is doing, and why. As yet, machines do not posess that type of 'self awareness'.(5)

As an inventor of art therefore, the Art Engine stumbles on this block: it is confined to its rule-bound world. Truck is aware of this; in the manual, with respect to Hypercard, he says: "Art Engine is not nearly so freeform. It offers a particular, highly associative structure for information that, when adhered to, gives ArtEngine the potential to behave intelligently." (6) In its defence, it should be remembered that the level of intelligent behavior that a Mac2 is capable of, in terms of a loose neurological comparison, is about that of a worm or sea slug. That's pretty creative for a sea slug. Thus limitations are externally imposed, by the technology as well as its socio-ecomomic context.

Pioneers in this new medium take on several large challenges. The technology is too primitive to allow user friendly realization of the project. A new audience must be educated to understand these limitations and be prepared to engage the skeleton of a work, at the expense of some effort.

More broadly this discussion poses the question: Does 'art' make a net gain or loss by engaging sophisticated, rapidly changing contemporary technologies. This depends on whether one views art as timeless, or bound to contemporary socio-economic structures. If one holds to the former, then clearly the employment of the technology is a net loss, if only because of the learning curve argument outlined above. But if one holds that position, one has to be prepared to ague that art does not change, and that Tang bronzes and Egyptian tombs address the same issues, in the same language, as Sherry Levine and Jeff Koons. This is clearly untenable. Thus both artists and viewers must accept the inevitability that computer based art will exist because the technology exists, and it will say things previously unsaid or even unsayable.

That said, the potential for computer based interactive artwork is limitless. The small but growing population of artists in the field continue to research the possibilities. Meantime the computer and entertainment industries are rapidly producing everything from interactive laserdiscs on Tibetan Thanka painting to Virtual Reality war simulations with you as Rambo. Interactive media is, and will remain for some time, the fastest growing educational and entertainment media. As a potential medium for artworks it is awesomely powerful.

POSTSCRIPT: Since writing this essay, I have received a new addition to the ArtEngine family: Illustrated ArtEngine, a 101 page artists' book by Fred Truck which traces the process of design of an artists' museum by ArtEngine, with copious commentary by the author. The focus of ArtEngines machinations is "museums made by artists, a genre that has a rich history in the 20th century. Works given include Disneyland, Disneyworld, Duchamps' Boite en valise, and Mouse Museum by Claes Oldenburg. And of course, its own critical synthesis of these: Engine-Mouse."(7) ArtEngine digests and synthesizes entire conceptual corpi: not simply intelligent image merging a la Nancy Burson, but interrelated bodies of image, text, active and passive data. ArtEngine achieves this extraordinary task by the application of a strict Hegelian dialectic "Artengine is...an electronic simulation of Hegelelian dialectical process, it is ideally positioned to encourage artists to collaborate with it in making art, because dialectics are a traditional part of art criticism. The challenge for artists is not to USE computers, but to COLLABORATE with them as they would another being, and to make art that by the fact of its existence synthesizes a new human/machine identity." (8)

I am continually defeated in my attempts to describe what ArtEngine is. *ArtEngine is an automated conceptual artwork* is partially satisfying. *An automated ego diffuser for human/machine collaborative art production* catches something of it. ArtEngine is something for which we have, as yet, no name. We may discuss ArtEngine in 20 years the way we currently discuss the Wright brothers.

ArtEngine is available from ArtCom/La Mamelle, San Francisco. Cost: $200
Illustrated ArtEngine is available from Art Com and Printed Matter, NYC. Cost: $15

1. ArtEngine manual p28
2. Harold Cohen, in a paper for the First International Symposium on Electronics in Art, Holland,1988.
3. I hold the opinion that some of the forms of conceptual or post object art constituted a kind of 'cultural software' that predated the availability of personal computers.

4. This situation is reminiscent of Goedels' Incompleteness theorum: '..in any sufficiently powerful logical system statements can be formulated which can neither be proved nor disproved within the system, unless possibly the system is inconsistent."(The minds I. Hofstadter and Dennett 1981 p58) A program (for instance, ArtEngine) is a set of mathematical rules within a proscribed universe. Godels' theorum indicates that a set of rules cannot describe its own universe. A very loose and generalized paraphrase might be that a program, by virtue of Godels' theorum, cannot have self-knowledge.
5. These ideas are well expounded in Hubert Dreyfus "What computers can't do. (revised edition)" Harper colophon books 1979. His critique is as relevant today as it was ten years ago.
6. ArtEngine manual p26
7. Illustrated ArtEngine p20
8. Illustrated ArtEngine p14